PAP

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/755,502	01/05/2001	Arthur H. Khu	X-779 US	5243

24309 7590 12/09/2003

XILINX, INC
ATTN: LEGAL DEPARTMENT
2100 LOGIC DR
SAN JOSE, CA 95124

EXAMINER	
CHOW, CHIH CHING	
ART UNIT	**PAPER NUMBER**
2122	2

DATE MAILED: 12/09/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

P12-9

Office Action Summary	Application No.	Applicant(s)	
	09/755,502	KHU, ARTHUR H.	
	Examiner	Art Unit	
	Chih-Ching Chow	2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01/15/2001* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *15 January 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☒ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

Claim Rejections - 35 USC § 112

1. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

2. Claim 8 is rejected under 35 U.S.C. 112 because it is not clear how does the claimed invention work, one skilled in the art clearly would not know how to use the claimed invention. Claim 8, "The method of optimizing as set forth in claim 1 wherein the **keyword statement is identified from a selection made by a user**." It's not clear how is the 'keyword statement' is identified from a selection made by a user? Normally a compiler has a built-in list of 'keyword statements' based on the programming language it is using. It's also stated in the Detailed Description of the Drawings [0018], "The source code 15 is a computer program written in programming language by a programmer, automatically generated by a computer..." there is no description in the claim neither in the specification about how the user is going to enter the selection of his/her own keyword statements.

3. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter, which the applicant regards as his invention.

4. Claim 8, "The method of optimizing as set forth in claim 1 wherein the **keyword statement is identified from a selection made by a user**." but, since the claim does not set forth any steps involved in the method/process, it is unclear what method/process applicant is intending to encompass. A claim is indefinite where it merely recites a use without any active, positive steps delimiting how this use is actually practiced. It's not clear how is the 'keyword statement' is identified from a selection made by a user? Feature in claim 8 is not clearly disclosed in the specification (see above).

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1-6, 8-17, 19 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Publication No. US2002/0087954 A1 by Wang et al. as applied to claims above, and further in view of U.S. Patent No. 5835771 by Veldhuizen.

The explanations for each of the rejected claims are as following:

Claims	Wang / Veldhuizen
1. A method of optimizing computer program code where the computer	Wang's invention has taught us a 'quantization' process, which "includes

program code includes a plurality of statements, the method comprising the steps of :

(1) identifying a keyword statement;

mapping n-dimensional vectors that correspond to instructions, live-in states, and live-out states to one dimensional symbols, and arranging the symbols into a text in program execution order." And a "discovery" process, which includes "**the identification** (as in claim 1 (1) *identifying a keyword statement*), of **recurrent symbols** and **recurrent phrases of symbols within the text** (as in claim 1 (2), *searching the program code for the keyword statement* and in claim 1 (3), *determining if the keyword statement begins a repeating pattern*). Recurrent symbols and phrases correspond to reuse instances" (See Wang's Abstract). Further more, in Wang's invention, page. 2, paragraph [0023] "The term 'discovery' refers to the process of **identifying**, or 'discovering' **recurrent portions** of the text. Discovering recurrent portions of text is equivalent to identifying reuse instances

	of single instruction and reuse instances of sequences of instructions." Here the "keyword statement" in claim 1 (1) has the same meaning as an "instruction", both meant an executable action in any programming language. Wang's invention goes through the entire program code and search for repeating pattern for all the instructions. The quantization process will be able to determine how many times the recurrence of a certain keyword is.
(2) searching the program code for the keyword statement;	See the rejection of claim 1 (1).
(3) determining if the keyword statement begins a repeating pattern of statements in the program code; and	See the rejection of claim 1(1).
(4) replacing the repeating pattern of statements with a program loop equivalent to the repeating pattern of statements.	Wang's invention is a method of finding repeated code. Wang does not specifically point out replacing the repeating code with a program loop. However, Veldhuizen's

invention has disclosed using a loop for repeated code. Veldhuizen shows using a loop for repeated code in an analogous art for the purpose of achieving greater efficiency of code. In Veldhuizen's invention has disclosed using a loop for repeated code, in column 1, 3rd paragraph, "A block of code that is **repeated 'n' number of times** is referred to as an **n-repetition loop**." A 'for' loop example is given in the same paragraph.
It would have been obvious to a person of the ordinary skill in the art at the time of the invention to modify Wang's system with the feature of replacing a repeated code by a loop for the same reason it is taught by Veldhuizen, to achieve greater efficiency by writing repeated program code in loops.

16. A process for optimizing a software code that includes a plurality of

In Wang's invention, page 2, paragraph [0020], "The entire execution of a program

statements, the process comprising the steps of:

(1) locating multiple occurrences of a code pattern within the software code where the **multiple occurrences** appear sequentially to each other in the software code;

can be represented as a trace of the aforementioned vectors, each corresponding to the execution of a single instruction. Some of the vectors are **unique**, and others are repeated **one or more times** in the trace. "

Wang's 'quantization' process actually includes **locating** the 'keyword', and **quantifying the occurrences of the instruction**, during the quantization process, it goes through the entire program text, perform the analysis of each of the program instruction, in Wang's [0022], "When quantization is complete, an entire execution trace of a program is represented in a text of symbols, each symbol corresponding to one execution of one instruction. Sequences of symbols within the text are referred to as 'phrases.'" Basically it includes adding subsequent non-keyword statement (data or operands for that

	instruction) to the instruction vector and comparing the consecutive code patterns.
(2) generating a program loop that executes one occurrence of the code pattern a number of times to produce an equivalent result as executing the multiple occurrences of the code pattern; and replacing the multiple occurrences of the code pattern in the software code with the program loop.	See the rejection of claim 1 (4).
19. The process for optimizing a software code as set for the in claim 16, prior to the locating step, further including:	For the features of claim 16 see Wang and Veldhuizen.
(1) selecting at least one keyword statement where the keyword statement includes a keyword and an optional data reference; and	Wang's page 1, paragraph [0016], "An execution trace of a program is represented by a sequence of multi-dimensional vectors, each vector corresponding to a dynamic **instance of an instruction (selecting keyword)** and its live-in states and live-out states

(**optional data reference**). The sequence of vectors is mapped into a text of one-dimensional symbols." The **instance of an instruction** corresponds the "**each keyword statement in the software code**", claim 19 (2). In Wang's page 2, [0019], "The **vector** are each represented by <instruction pointer (IP), live-in states, live-out states>. By default, **the live-in states and live-out states are source and destination operands**, respectively". A **conversion** of the data references is done here (so the data references appear in each keyword statement in the software code data array), the IP, the live-in and live-out values are **'optional data reference'**. The one-dimensional symbols (vector) correspond to the **data array reference.**

(2) converting each of the data references that appear in each keyword

See the rejection of 19 (1).

statement in the software code to a data array reference, the data array reference being loaded with values of the converted data references.

2. The Method of optimizing a set forth in claim 1 wherein the keyword statement includes a keyword and an optional data references, the method further including prior to the searching step:

For the features of claim 1 see Wang and Veldhuizen.

Sequentially locating each keyword statement in the program code; and

See the rejection of claim 1 (1) and 16 (1).

Converting the optional data reference, if present, from each located keyword statement to a data array references.

See the rejection of claim 19 (1).

3. The method of optimizing as set forth in claim 2 wherein the converting includes assigning an array index value to the data

For the features of claim 2 see Wang and Veldhuizen.

In Wang's invention, page 8, claim 3:

array reference where each located keyword statement is assigned a next sequential value of the array index value.

"assigning new symbols comprises assigning consecutive integers such that each new symbol is assigned a value that is one greater than a previously assigned value", here Wang is constructing a vector which comprised by found keywords, the occurrences of each of the keywords will be one of the attributes of the array.

Also in page 3, paragraph [0037]

"Instructions in the execution trace are labeled as vectors to indicate that each executed instruction is represented by <IP, live-in states, live-out states>. In this example, symbols are assigned to vectors such that each new vector is assigned the **next available integer**." The first symbol (the integer zero) is assigned to vector 302, which corresponds to instruction 206, and the second symbol (the integer "one") is assigned to vector 304." The array index assignment is a basic computer programming skill; the key concept is to

	assign the next sequential value as the next index number for a newly identified array item.
5. The method of optimizing as set for the in claim 1 wherein the determining step includes:	For the features of claim 1 see Wang and Veldhuizen.
determining a first pattern of statements in the program code beginning with a first keyword statement and ending with a statement preceding a second keyword statement that sequentially appears in the program code after the first keyword statement;	See the rejection of claim 1. All the 'determining keyword statement' is part of the 'identifying', 'searching' and 'determining' recited in claim 1.
determining a second pattern of statements in the program code beginning with the second keyword statement and ending with a statement preceding a third keyword statement that sequentially appears in the program code after the second keyword statement; and	See the rejection of claim 1.

comparing the first pattern of statements to the second pattern of the statements; and	See the rejection of claims 1 and 16 (1).
setting the first pattern of statements as a repeating pattern if the first and second pattern of statements substantially match.	See the rejection of claims 1 and 16 (1).

6. The method of optimizing as set forth in claim 1 wherein the replacing step includes:	For the features of claim 1 see Wang and Veldhuizen.
generating loop code for executing a loop within the source code at a location of the repeating pattern of statements;	See the rejection of claim 1 (4).
inserting one instance of the repeating pattern of statements within the loop code; and	See the rejection of claim 1 (4).
defining the loop code to iterate a number of times equal to a number of instances of the repeating pattern.	See the rejection of claim 1 (4).

8. The method of optimizing as set forth in claim 1 wherein the keyword statement is identified from a selection made by a user.	For the features of claim 1 see Wang and Veldhuizen. This claim is rejected as its parent claim. For the feature of claim 8 see the rejection of 35 U.S.C. 112.
9. The method of optimizing as set forth in claim 1 further including identifying a plurality of keyword statements and repeating the method for optimizing for each of the plurality of keyword statements.	For the features of claim 1 see Wang and Veldhuizen. For the rest of the claim see the rejection of claims 1 and 16 (1).
10. A software code optimizer comprising:	
(1) analyzing program instructions for analyzing a software code and determining an occurrence of a repeating pattern of code therein; and	See the rejection of claim 1 (1); the discovering process is the same as 'analyzing' as recited in Claim 10 (1).
(2) converting program instructions for converting the repeating pattern of code for a programming loop that performs an equivalent function as the repeating	See the rejection of claim 1 (4).

pattern of code.

11. The software code optimizer as set forth in claim 10 wherein the analyzing program instructions further include:

program instructions for searching the software code for a keyword; and

For the features of claim 10 see Wang and Veldhuizen.

See the rejection of claim 1 (1).

program instructions for identifying if the keyword begins a repeating pattern of code within the software code and determining a number of occurrences of the repeating pattern.

See the rejection of claim 1 (1).

13. The software code optimizer as set forth in claim 11 wherein the converting program instructions further include program instructions for setting the programming loop to repeat a number of times equal to the number of occurrences of the repeating pattern and inserting one occurrence of the repeating pattern within

For the features of claim 11 see Wang and Veldhuizen. For the rest of the claim see the rejection of claim 1(4).

the programming loop.

17. The process for optimizing a software code as set forth in claim 16 further including:	For the features of claim 16 see Wang and Veldhuizen.
selecting a keyword statement;	See the rejection of claim 1 (1).
defining the code pattern based on the keyword statement.	See the rejection of claim 1 (1).
20. The process for optimizing a software code as set forth in claim 16 wherein the generating a program loop step includes generating a looping instruction.	For the features of claim 16 see Wang and Veldhuizen. For the rest of the claim see the rejection of claim 1 (1).
14. The software code optimizer as set forth in claim 10 further including program instructions for locating data references in each statement of program code containing the keyword and converting the data references to data array references.	For the features of claim 10 see Wang and Veldhuizen. For the rest of the claim see the rejection of claim 1, 16(1) and 19 (1).

4. The method of optimizing as set forth in claim 3 wherein the determining step further includes:	For the features of claim 3 see Wang and Veldhuizen.
comparing data array references of two keyword statements from the program code; and	See the rejection of claim 16 (1).
determining if the array index values from the data array references match in size and sequential order.	See the rejection of claim 3.
12. The software code optimizer as set forth in claim 11 further including program instructions for repeating the analyzing program instructions for a plurality of keywords.	For the features of claim 11 see Wang and Veldhuizen. For the rest of the claim see the rejection of claim 9.
15. The software code optimizer as set forth in claim 10 further including a compiler for translating the software code to an object code executable by a computer.	For the features of claim 10 see Wang and Veldhuizen. Wang's invention teaches the method of building up a keyword array associate with

program optional data, but does not teach specifically using a compiler to translate the software code to an object code; however Veldhuizen shows using a compiler to translate software code to an object code executable by a computer in an analogous art for the purpose of to avoid compiling errors. In Veldhuizen's disclosure, column 1, item 2, "Background Art", "Before a software application can be executed on a computer system, it is expressed in a programming language. A programming language can be assembly language, for example, or a higher-level language such as Basic, C, or Pascal. The expression of a software application in a programming language is referred to as code. The code expresses a flow of execution for the operations that are performed in the software application. For example, the code expresses the flow of execution for retrieving input form a user of

the application. **Once the code is written, it is translated into object code using a compiler**." Same as recited in claim 15. It would have been obvious to a person of ordinary skill in the art at the time of the invention to modify Wang's method of using compiler to translate software code into object code for the same reason it is taught by Veldhuizen, to use a compiler to generate object code in order for the compiler to perform the actions specified in the program instructions.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claims 7 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Publication No. US2002/0087954 A1 by Wang et al. as applied to claims above, further

in view of U.S. Patent No. 5835771 by Veldhuizen, and further in view of the "Microsoft Press Computer Dictionary" (MPCD herein after), 1997.

The explanations for each of the rejected claims are as following:

Claims	Wang / Veldhuizen / MPCD
7. The method of optimizing as set forth in claim 1 wherein the keyword statement is identified from a predetermined keyword statement.	For the features of claim 1 see Wang and Veldhuizen. Wang's invention teaches the method of building up a keyword array associate with program optional data. Veldhuizen shows to convert repeated code into a loop. But neither of them specifically said the keyword statement is 'predefined'. However, in Microsoft Computer Dictionary, **Statement** is defined as: the smallest executable entity within a *programming language*. **Keyword** is defined as: Any of the set of words that composes a given *programming language* of set of operating system routines. A keyword statement is identified from a "predetermined" or "predefined" set so the compiler can recognize it and the O.S. to

	execute it. It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to modify Wang's invention with the pre-defined "keyword statement" for computer programming work, for the same reason it is taught by Microsoft Press Computer Dictionary, to avoid compiling errors.
18. The process for optimizing a software code as set forth in claim 7 wherein the defining step includes:	For the features of claim 7 see Wang, Veldhuizen, and MPCD.
locating a first instance of the keyword statement in the software code;	See the rejection of claims 1(1) and 16 (1).
defining a first code pattern to include at least the first instance of the keyword statement;	See the rejection of claim 1 (1).
adding subsequent non-keyword statement to the first code pattern until a second instance of the keyword statement	See the rejection of claim 16 (1).

appears I the software code;

defining a second code pattern to include at least the second instance of the keyword statement;	See the rejection of claim 1 (1).
adding subsequent non-keyword statements to the second code pattern until a third instance of the keyword statement appears in the software code or until a number of the subsequent non-keyword statements added equal a number of the subsequent non-keyword statements in the fist code pattern; and	See the rejection of claim 1 (1).
comparing the first code pattern with the second code pattern to determine if the second code pattern is a multiple occurrence of the first code pattern.	See the rejection of claims 1 and 16 (1).

Conclusion

9. The following summarizes the status of all the claims:

112 1st paragraph rejection: 8

112 2nd paragraph rejection: 8

103 rejections: 1 - 20

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chih-Ching Chow whose telephone number is 703-305-7205. The examiner can normally be reached on 6:30am to 3:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached on. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 703-305-3900.

Chih-Ching Chow
Examiner
Art Unit 2122

CC

JOHN CHAVIS
PATENT EXAMINER
ART UNIT 2124